Amendments to the Claims:

This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims:

1. (Currently Amended) A computer-implemented method for computing the number of points on an elliptic curve ~~over a finite field, in which a Frobenius equation is solved to a given precision by first and second parts, wherein said parts comprise the following steps~~, the method comprising:

receiving an elliptic curve having a ~~total~~ number of points on the ~~entire~~ curve; ~~and~~

determining, with a processor, the ~~total~~ number of points on the elliptic curve, wherein the determining includes solving a lifted Frobenius equation to a full precision by ~~using first and second parts with a reduced precision~~ computing a plurality of partial solutions at a plurality of successively reduced precisions, wherein the solving includes:

(a) computing, to ~~the~~ a first reduced precision, a first partial solution of said lifted Frobenius equation ~~using said first part recursively~~,

(b) applying a Frobenius operation to said first partial solution,

(c) computing an error term for said lifted Frobenius equation using the first partial solution and/or a result of step (b),

(d) computing correction factors for said lifted Frobenius equation using the first partial solution and/or a result of step (b),

(e) computing, to the first reduced precision, a second partial solution of a modified lifted Frobenius equation ~~that includes~~ using the error term ~~and the correction factors using said second part~~, wherein computing the second partial solution includes:

(1) computing, to another reduced precision, a third partial solution of said modified lifted Frobenius equation ~~using said second part~~ by recursively performing steps (1)-(5) to solve said modified lifted Frobenius equation from a lowest reduced precision to the another reduced precision, wherein the another reduced precision is less than the first reduced precision,

(2) applying a Frobenius operation to said third partial solution,

(3) updating said error term using results of steps (1) and (2) and the correction factors,

(4) computing, to the another reduced precision, a fourth partial solution of said modified lifted Frobenius equation with the updated error term ~~using said second part~~ by recursively performing steps (1)-(5) to solve said modified lifted Frobenius equation with the updated error term from a lowest reduced precision to the another reduced precision, and

(5) combining said third partial solution and said fourth partial solution to create the second partial solution,

(f) combining said first partial solution and said second partial solution ~~to provide the solution at the full precision~~; and

(g) repeating steps (a)-(f) one or more additional times to solve the lifted Frobenius equation to a full precision, wherein the result from step (f) is used as the first partial solution of step (a) for the next successively higher precision; and

based on the number of points on the elliptic curve, generating a cryptographic key for use in a digital processing system.

2. (Currently Amended) The method of claim 1 in which each successive precision ~~said reduced precision~~ is one half of ~~said full~~ the previous precision.

3. (Currently Amended) The method of claim 1 in which steps (a)-(f) and (1)-(5) ~~said first and second parts~~ compute the Teichmüller lift of a given finite-field polynomial.

4. (Currently Amended) The method of claim 1 in which steps (a)-(f) and (1)-(5) ~~said first and second parts~~ compute the canonical lift of said elliptic curve.

5. (Currently Amended) The method of claim 1 in which steps (a)-(f) and (1)-(5) ~~said first and second parts~~ compute the multiplicative representative of a given finite-field element.

6. (Currently Amended) The method of claim 1 in which steps (a)-(f) and (1)-(5) ~~said first and second parts~~ compute the trace of a given p-adic number.

7. (Currently Amended) The method of claim 1 in which steps (a)-(f) and (1)-(5) ~~said first and second parts~~ compute the norm of a given p-adic number.

8. (Currently Amended) The method of claim ~~1~~0 1, further comprising:

receiving a sequence of elliptic curves and determining the ~~total~~ number of points on each elliptic curve~~, in which said first and second parts analyze the sequence of elliptic curves~~.

9. (Currently Amended) The method of claim 8, further comprising:

generating a cryptographic key for use in a digital processing system using one of the secure elliptic curves based on the values for the number of points for the elliptic curves.

10. (Currently Amended) The method of claim 1, further comprising:

based on the ~~total~~ number of points, identifying whether the elliptic curve is a secure elliptic curve for generating a cryptographic key.

11. (Currently Amended) The method of claim 1, further comprising:

storing the ~~total~~ number of points on the elliptic curve in a memory of the computer.

12. (Currently Amended) A computer readable medium embodying program code for ~~directing~~ executing by one or more processors to perform an operation for computing the number of points on an elliptic curve, the operation comprising ~~the steps of~~:

receiving an elliptic curve having a ~~total~~ number of points on the ~~entire~~ curve; ~~and~~

determining the ~~total~~ number of points on the elliptic curve, wherein the determining includes solving a lifted Frobenius equation to a full precision by ~~using first and second parts with a reduced precision~~ computing a plurality of partial solutions at a plurality of successively reduced precisions, wherein the solving includes:

(a) computing, to ~~the~~ a first reduced precision, a first partial solution of said lifted Frobenius equation ~~using said first part recursively~~,

(b) applying a Frobenius operation to said first partial solution,

(c) computing an error term for said lifted Frobenius equation using the first partial solution and/or a result of step (b),

(d) computing correction factors for said lifted Frobenius equation using the first partial solution and/or a result of step (b),

(e) computing, to the first reduced precision, a second partial solution of a modified lifted Frobenius equation ~~that includes~~ using the error term ~~and the correction factors using said second part~~, wherein computing the second partial solution includes:

(1) computing, to another reduced precision, a third partial solution of said modified lifted Frobenius equation ~~using said second part~~ by recursively performing steps (1)-(5) to solve said modified lifted Frobenius equation from a lowest reduced precision to the another reduced precision, wherein the another reduced precision is less than the first reduced precision,

(2) applying a Frobenius operation to said third partial solution,

(3) updating said error term using results of steps (1) and (2) and the correction factors,

(4) computing, to the another reduced precision, a fourth partial solution of said modified lifted Frobenius equation with the updated error term ~~using said second part~~ by recursively performing steps (1)-(5) to solve said modified lifted Frobenius equation with the updated error term from a lowest reduced precision to the another reduced precision, and

(5) combining said third partial solution and said fourth partial solution to create the second partial solution,

(f) combining said first partial solution and said second partial solution ~~to provide the solution at the full precision~~; and

(g) repeating steps (a)-(f) one or more additional times to solve the lifted Frobenius equation to a full precision, wherein the result from step (f) is used as the first partial solution of step (a) for the next successively higher precision; and

based on the number of points on the elliptic curve, generating a cryptographic key for use in a digital processing system.

13. (Currently Amended) The computer readable medium of claim 12, wherein the operation further comprises ~~the step of~~:

based on the ~~total~~ number of points, identifying whether the elliptic curve is a secure elliptic curve for generating a cryptographic key.

14. (Currently Amended) ~~A~~ An integrated circuit configured to compute the number of points on an elliptic curve, the integrated circuit comprising:

hardware logic that receives an elliptic curve having a ~~total~~ number of points on the ~~entire~~ curve;

hardware logic that determines the ~~total~~ number of points on the ~~elliptic~~ curve, wherein the determining includes solving a lifted Frobenius equation to a full precision by ~~using first and second parts with a reduced precision~~ computing a plurality of partial solutions at a plurality of successively reduced precisions, wherein the solving includes:

(a) computing, to ~~the~~ a first reduced precision, a first partial solution of said lifted Frobenius equation ~~using said first part recursively~~,

(b) applying a Frobenius operation to said first partial solution,

(c) computing an error term for said lifted Frobenius equation using the first partial solution and/or a result of step (b),

(d) computing correction factors for said lifted Frobenius equation using the first partial solution and/or a result of step (b),

(e) computing, to the first reduced precision, a second partial solution of a modified lifted Frobenius equation ~~that includes~~ using the error term ~~and the correction factors using said second part~~, wherein computing the second partial solution includes:

(1) computing, to another reduced precision, a third partial solution of said modified lifted Frobenius equation ~~using said second part~~ by recursively performing steps (1)-(5)

to solve said modified lifted Frobenius equation from a lowest reduced precision to the another reduced precision, wherein the another reduced precision is less than the first reduced precision;

(2) applying a Frobenius operation to said third partial solution,

(3) updating said error term using results of steps (1) and (2) and the correction factors,

(4) computing, to the another reduced precision, a fourth partial solution of said modified lifted Frobenius equation with the updated error term ~~using said second part~~ by recursively performing steps (1)-(5) to solve said modified lifted Frobenius equation with the updated error term from a lowest reduced precision to the another reduced precision, and

(5) combining said third partial solution and said fourth partial solution to create the second partial solution,

(f) combining said first partial solution and said second partial solution ~~to provide the solution at the full precision~~; and

(g) repeating steps (a)-(f) one or more additional times to solve the lifted Frobenius equation to a full precision, wherein the result from step (f) is used as the first partial solution of step (a) for the next successively higher precision; and

hardware logic that based on the number of points on the elliptic curve generates a cryptographic key for use in a digital processing system.

15. (Currently Amended) The integrated circuit of claim 14, further comprising:

hardware logic for identifying, based on the ~~total~~ number of points, the elliptic curve as a secure elliptic curve for generating a cryptographic key.

16. (New) The computer readable medium of claim 12 wherein each successive precision is one half of the previous precision.

17. (New) The computer readable medium of claim 12, wherein the operation further comprises:

receiving a sequence of elliptic curves and determining the number of points on each elliptic curve.

18. (New) The computer readable medium of claim 17, wherein the operation further comprises:

generating a cryptographic key for use in a digital processing system using one of the secure elliptic curves based on the values for the number of points for the elliptic curves.

19. (New) The integrated circuit of claim 14, wherein each successive precision is one half of the previous precision.

20. (New) The integrated circuit of claim 14, wherein the hardware logic receives a sequence of elliptic curves and determines the number of points on each elliptic curve.

21. (New) The integrated circuit of claim 14, further comprising:

hardware logic that generates a cryptographic key for use in a digital processing system using one of the secure elliptic curves based on the values for the number of points for the elliptic curves.